Turingovy stroje a jazyky

TM $\mathcal{M}$ **akceptuje**/**rozpoznává**/**přijímá** jazyk $L(\mathcal{M})$.
Jazyk $L(\mathcal{M})$ je **rekursivně spočetný**.

Je-li TM $\mathcal{M}$ úplný, říkáme, že $\mathcal{M}$ **rozhoduje** jazyk $L(\mathcal{M})$.
Jazyk $L(M)$ je **rekursivní**.

Věta. Jazyk ke akceptovaný Turingovým strojem, právě když je generovaný nějakou (frázovou) gramatikou.

Důkaz. Neuveden. $\square$

Uzávěrové vlastnosti rekursivních a rekursivně spočetných jazyků

Věta. Rekursivně spočetné i rekursivní jazyky jsou uzavřené na **sjednocení**.

Důkaz.

Nechť L_1, L_2 jsou jazyky akceptované TM $\mathcal{M}_1, \mathcal{M}_2$ s disjunktními množinami stavů. Stroj $\mathcal{M}$ akceptující $L_1 \cup L_2$ se nedeterministicky rozhodne, zda na svém vstupu spustí $\mathcal{M}_1$ nebo $\mathcal{M}_2$. Pokud spuštěný stroj akceptuje nebo zamítne, stroj $\mathcal{M}$ udělá totéž. Pokud $\mathcal{M}_1, \mathcal{M}_2$ byly úplné, je i $\mathcal{M}$ úplný. $\square$

Uzávěrové vlastnosti rekursivních a rekursivně spočetných jazyků

Věta. Rekursivně spočetné i rekursivní jazyky jsou uzavřené na **průnik**.

Důkaz.

Nechť L_1, L_2 jsou jazyky akceptované TM $\mathcal{M}_1, \mathcal{M}_2$ s disjunktními množinami stavů. Stroj $\mathcal{M}$ akceptující $L_1 \cap L_2$ spustí na svém vstupu w nejprve $\mathcal{M}_1$ a pokud akceptuje, pak na w spustí i $\mathcal{M}_2$. $\mathcal{M}$ akceptuje jen pokud oba stroje akceptují. Pokud $\mathcal{M}_1, \mathcal{M}_2$ byly úplné, je i $\mathcal{M}$ úplný. $\square$

Uzávěrové vlastnosti rekursivních a rekursivně spočetných jazyků

Věta. Rekursivně spočetné i rekursivní jazyky jsou uzavřené na **zřetězení**.

Důkaz.

Nechť L_1, L_2 jsou jazyky akceptované TM $\mathcal{M}_1, \mathcal{M}_2$ s disjunktními množinami stavů. Dvojpáskový stroj $\mathcal{M}$ akceptující $L_1.L_2$ vstupní slovo nedeterministicky rozdělí na dvě části, každou část dá na jednu pásku). Na první části slova spustí $\mathcal{M}_1$. Pokud akceptuje, spustí na druhé části $\mathcal{M}_2$. $\mathcal{M}$ akceptuje jen pokud oba stroje akceptují. Pokud $\mathcal{M}_1, \mathcal{M}_2$ byly úplné, je i $\mathcal{M}$ úplný. $\square$

Uzávěrové vlastnosti rekursivních a rekursivně spočetných jazyků

Věta. Rekursivně spočetné i rekursivní jazyky jsou uzavřené na **iteraci**.

Důkaz.

Nechť L_1 je jazyky akceptovaný TM $\mathcal{M}_1$. Zkonstruujeme dvojpáskový TM $\mathcal{M}$ rozpoznávající L_1^*. $\mathcal{M}$ akceptuje, pokud je na první pásce prázdné slovo. V opačném případě nedeterministicky přesune nějaký neprázdný prefix slova z první pásky na druhou pásku, kde nad ním spustí $\mathcal{M}_1$. Pokud $\mathcal{M}_1$ zamítne, tak i $\mathcal{M}$ zamítne. Jinak opakuje celou proceduru. Je-li $\mathcal{M}_1$ úplný, je i $\mathcal{M}$ úplný. $\square$

Uzávěrové vlastnosti rekursivních a rekursivně spočetných jazyků

Věta. Rekursivní jazyky jsou uzavřené na **doplněk**.

Důkaz.

Nechť L_1 je jazyky akceptovaný úplným TM $\mathcal{M}_1$. Úplný stroj $\mathcal{M}$ rozhodující $co{-}L_1$ získáme záměnou akceptujícího a zamítajícího stavu. $\square$

Uzávěrové vlastnosti rekursivních a rekursivně spočetných jazyků

Věta. Třída rekursivně spočetných jazyků **není** uzavřená na **doplněk**.

Důkaz. Provedeme později. $\square$

Třída rekursivně spočetných jazyků **je uzavřená** na sjednocení, průnik, zřetězení, iteraci, pozitivní iteraci, mocniny.

Třída rekursivních jazyků **je uzavřená** na sjednocení, průnik, zřetězení, iteraci, pozitivní iteraci, mocniny, doplněk.

Vztah rekursivních a rekursivně spočetných jazyků

Věta. Jazyk L je rekursivní, právě když jsou jazyky L a $co{-}L$ rekursivně spočetné.

Důkaz. "$\Longrightarrow$" plyne z uzavřenosti rekursivních jazyků na komplement a z toho, že každý rekursivní jazyk je i rekursivně spočetný.

"$\Longleftarrow$": Nechť $\mathcal{M}_1, \mathcal{M}_2$ jsou TM rozponávající L a $co{-}L$. Sestrojíme dvojpáskový stroj $\mathcal{M}$ rozhodující L. $\mathcal{M}$ nejprve zkopíruje vstupní slovo w na druhou pásku. Pak paralelně vykonává běh $\mathcal{M}_1$ na první pásce a běh stroje $\mathcal{M}_2$ na druhé pásce. Pokud stroj $\mathcal{M}_1$ akceptuje, pak $w \in L$ a stroj $\mathcal{M}$ také akceptuje. Pokud stroj $\mathcal{M}_2$ akceptuje, pak $w \notin L$ a stroj $\mathcal{M}$ zamítá. $\square$

Problémy jako jazyky

Problém rozhodnout, zda daný řetězec w má vlastnost P lze ztotožnit s množinou $\{w \mid w \text{ má vlastnost } P\}$.

Objekty O lze kódovat jako slova $\langle O \rangle$. Problém, zda O má vlastnost P ztotožníme s jazykem $\{\langle O \rangle \mid O \text{ má vlastnost } P\}$.

Příklad. Problém rozhodnout, zda daný konečný graf je souvislý, ztotožníme s jazykem $\{\langle G \rangle \mid G \text{ je konečný souvislý graf}\}$.

Kódování TM

Každý TM lze zakódovat do binárního řetězce. Předpokládáme, že $\mathcal{M}$

- má stavy $Q = \{q_1, q_2, q_3, \ldots, q_n\}$
- q_1 je iniciální stav, q_2 akceptující stav, q_3 zamítající stav
- má páskovou abecedu $\Gamma = \{X_1, X_2, \ldots, X_z\}$
- $X_1 = \triangleright$ je levá koncová značka, $X_2 = \sqcup$ je symbol pro prázdné pole

Přechod $\delta(q_i, X_j) = (q_k, X_l, L)$ kódujeme řetězcem $0^i10^j10^k10^l10$.
Přechod $\delta(q_i, X_j) = (q_k, X_l, R)$ kódujeme řetězcem $0^i10^j10^k10^l100$.

Z kódů jednotlivých přechodů (v libovolném pořadí) sestavíme kód $\mathcal{M}$:

$$\langle \mathcal{M} \rangle = 111\, \textit{kód}_1\, 11\, \textit{kód}_2\, 11 \ldots 11\, \textit{kód}_r\, 111$$

Řetězce nekódující žádný TM považujeme za kód TM akceptujícího $\emptyset$.

Slova kódujeme podobně. Celkem $\langle \mathcal{M}, w \rangle = \langle \mathcal{M} \rangle \# \langle w \rangle$.

Univerzální Turingův stroj

Věta. Existuje **univerzální Turingův stroj** $\mathcal{U}$, který dokáže simulovat libovolný zadaný TM na zadaném vstupu:

$$\mathcal{U} \text{ akceptuje } \langle \mathcal{M} \rangle \# \langle w \rangle \iff \mathcal{M} \text{ akceptuje } w$$

Důkaz.

Stroj $\mathcal{U}$ je třípáskový. Nejprve ověří, že vstup je tvaru $\{0,1\}^*\#\{0,1\}^*$ (pokud ne, zamítá). Na první pásce je kód simulovaného stroje $\mathcal{M}$, na druhé pásce probíhá jeho výpočet, na třetí je uložen jeho stav. V každém kroku se na základě simulovaného stavu a obsahu pásky najde na první pásce příslušný přechod, který se pak provede. $\square$

Rozhodnutelnost problémů

Definice. Problém P odpovídající jazyku $L = \{\langle O \rangle \mid O \text{ má vlastnost } P\}$ je

- **rozhodnutelný**, právě když L je rekursivní
- **nerozhodnutelný**, právě když L je není rekursivní
- **částečně rozhodnutelný (semirozhodnutelný)**, právě když L je rekursivně spočetný

Problém akceptování

Problém akceptování (problém příslušnosti pro Turingovy stroje) je problém rozhodnout, zda daný TM $\mathcal{M}$ akceptuje dané slovo w nad jeho vstupní abecedou. Problém ztotožníme s jazykem

$$ACC = \{\langle \mathcal{M}, w \rangle \mid \mathcal{M} \text{ je TM a } \mathcal{M} \text{ akceptuje } w\}.$$

Věta. Problém akceptování je částečně rozhodnutelný.

Důkaz. Plyne z existence univerzálního Turingova stroje. $\square$

Věta. Problém akceptování je nerozhodnutelný.

Důkaz. Předpokládejme, že existuje TM $\mathcal{A}$ rozhodující problém akceptování. Tedy $\mathcal{A}$ akceptuje $\langle \mathcal{M}, w \rangle$, právě když $\mathcal{M}$ akceptuje w.

S využitím $\mathcal{A}$ zkonstruujeme TM $\mathcal{D}$: dostane-li $\mathcal{D}$ na vstupu zakódovaný stroj $\langle \mathcal{M} \rangle$, zeptá se stroje $\mathcal{A}$, zda $\mathcal{M}$ akceptuje svůj vlastní kód $\langle \mathcal{M} \rangle$ a následně odpověď otočí. Tedy

$$\begin{array}{lll} \mathcal{D} \text{ akceptuje } \langle \mathcal{M} \rangle, & \text{pokud} & \mathcal{M} \text{ neakceptuje } \langle \mathcal{M} \rangle \text{ a} \\ \mathcal{D} \text{ neakceptuje } \langle \mathcal{M} \rangle, & \text{pokud} & \mathcal{M} \text{ akceptuje } \langle \mathcal{M} \rangle. \end{array}$$

Nyní spustíme $\mathcal{D}$ na vstupu $\langle \mathcal{D} \rangle$:

$$\begin{array}{lll} \mathcal{D} \text{ akceptuje } \langle \mathcal{D} \rangle, & \text{pokud} & \mathcal{D} \text{ neakceptuje } \langle \mathcal{D} \rangle \text{ a} \\ \mathcal{D} \text{ neakceptuje } \langle \mathcal{D} \rangle, & \text{pokud} & \mathcal{D} \text{ akceptuje } \langle \mathcal{D} \rangle. \end{array}$$

To je spor. Stroj $\mathcal{A}$ tedy neexistuje a problém akceptování je nerozhodnutelný. $\square$

Ne-semirozhodnutelné problémy

Věta. Doplněk problému akceptování není ani částečně rozhodnutelný, tedy $co{-}ACC$ není rekursivně spočetný.

Důkaz.

$\square$

Problém zastavení

Problém zastavení (halting problem) je problém rozhodnout, zda daný TM $\mathcal{M}$ má na daném slově w nad jeho vstupní abecedou konečný výpočet (tedy zda $\mathcal{M}$ na vstupu w zastaví). Problém ztotožníme s jazykem

$$HALT = \{\langle \mathcal{M}, w \rangle \mid \mathcal{M} \text{ je TM a výpočet } \mathcal{M} \text{ na } w \text{ je konečný}\}.$$

Věta. Problém zastavení je částečně rozhodnutelný.

Důkaz. Pomocí univerzálního Turingova stroje simulujeme $\mathcal{M}$ na w. Pokud simulovaný výpočet skončí, akceptujeme. $\square$

Věta. Problém zastavení je nerozhodnutelný.

Důkaz. Předpokládejme, že existuje úplný TM $\mathcal{H}$ rozhodující problém zastavení. Pak ovšem umíme sestrojit TM $\mathcal{A}$ rozhodující problém akceptování. Stroj $\mathcal{A}$ dekóduje dvojici $\langle \mathcal{M}, w \rangle$ ze vstupu a změní $\mathcal{M}$ tak, že místo přechodů do zamítajícího stavu začne cyklit. Modifikovaný stroj $\mathcal{M}'$ zastaví, právě když $\mathcal{M}$ akceptuje. Nyní stačí spustit $\mathcal{H}$ na vstupu $\langle \mathcal{M}', w \rangle$. Dostáváme tedy úplný TM $\mathcal{A}$ rozhodující problém akceptování. To je spor. Úplný stroj $\mathcal{H}$ tedy neexistuje. $\square$

Redukce - Intuice

Turingovy stroje a funkce

Turingův stroj může reprezentovat i funkci. Výsledek je na výstupní pásce při akceptování. Pokud stroj $\mathcal{M}$ na vstupu w zastaví s obsahem pásky $\triangleright y \sqcup^\omega$ (kde y nekončí na $\sqcup$), pak y označíme jako $\mathcal{M}(w)$.

$f : \Sigma^* \to \Phi^*$

$g : \mathbb{N}_0^k \to \mathbb{N}_0$

Vyčíslitelné funkce

Definice. Funkce $f : \Sigma^* \to \Phi^*$ je **vyčíslitelná**, pokud existuje TM $\mathcal{M}$, který na vstupu w zastaví, právě když $f(w)$ je definovaná a navíc $f(w) = \mathcal{M}(w)$.

Funkce je **totálně vyčíslitelná**, pokud je vyčíslitelná a totální.

Redukce

Definice. Nechť $A \subseteq \Sigma^*$ a $B \subseteq \Phi^*$ jsou jazyky. Řekneme, že A se **m-redukuje** na B, píšeme $A \leq_{\mathrm{m}} B$, právě když existuje totálně vyčíslitelná funkce $f : \Sigma^* \to \Phi^*$ taková, že

$$w \in A \iff f(w) \in B.$$

Funkci f nazveme **redukcí** A na B.

A a B jsou **m-ekvivalentní**, psáno $A \equiv_{\mathrm{m}} B$, pokud $A \leq_{\mathrm{m}} B$ a $B \leq_{\mathrm{m}} A$.

Redukce a rozhodnutelnost

Věta. Nechť $A \leq_m B$.

- B je rekursivní $\implies$ A je rekursivní.
- B je rekursivně spočetný $\implies$ A je rekursivně spočetný.

Důkaz.

Nechť f je redukce A na B a $\mathcal{M}_B$ je TM akceptující B.
Stroj $\mathcal{M}_A$ akceptující A na vstupu w

1. spočítá $f(w)$
2. spustí $\mathcal{M}_B$ na vstupu $f(w)$ a vrátí stejný výsledek jako $\mathcal{M}_B$

Je-li $\mathcal{M}_B$ úplný, pak je i $\mathcal{M}_A$ úplný. $\square$

Redukce a rozhodnutelnost

Důsledek. Nechť $A \leq_{\mathrm{m}} B$.

- A není rekursivní $\implies$ B není rekursivní.
- A není rekursivně spočetný $\implies$ B není rekursivně spočetný.

Důsledek. Nechť $A \equiv_{\mathrm{m}} B$.

- A je rekursivní $\iff$ B je rekursivní.
- A je rekurisvně spočetný $\iff$ B je rekursivně spočetný.

$HALT \equiv_{\mathbf{m}} ACC$

$$\begin{aligned} HALT &= \{\langle \mathcal{M}, w\rangle \mid \mathcal{M} \text{ je TM a výpočet } \mathcal{M} \text{ na } w \text{ je konečný}\} \\ ACC &= \{\langle \mathcal{M}, w\rangle \mid \mathcal{M} \text{ je TM a } \mathcal{M} \text{ akceptuje } w\} \end{aligned}$$

$HALT \leq_{\mathrm{m}} ACC$:

$\mathit{HALT} \equiv_{\mathbf{m}} \mathit{ACC}$

$$\begin{aligned} \mathit{HALT} &= \{\langle \mathcal{M}, w \rangle \mid \mathcal{M} \text{ je TM a výpočet } \mathcal{M} \text{ na } w \text{ je konečný}\} \\ \mathit{ACC} &= \{\langle \mathcal{M}, w \rangle \mid \mathcal{M} \text{ je TM a } \mathcal{M} \text{ akceptuje } w\} \end{aligned}$$

$\mathit{ACC} \leq_{\mathrm{m}} \mathit{HALT}$:

Problém neprázdnosti

Problém neprázdnosti je problém rozhodnout, zda daný TM akceptuje neprázdný jazyk.

$$\mathit{NONEMPTY} = \{\langle \mathcal{M} \rangle \mid \mathcal{M} \text{ je TM a } L(\mathcal{M}) \neq \emptyset\}$$

Věta. Problém neprázdnosti není rozhodnutelný.

Důkaz. $\mathit{ACC} \leq_{\mathrm{m}} \mathit{NONEMPTY}$:

$\square$

Postův systém

Definice. Postův systém P nad ebecedou Σ je konečná množina dvojic

$$P = \{(\alpha_i, \beta_i) \mid \alpha_i, \beta_i \in \Sigma^*, 1 \leq i \leq n\}.$$

Řešením systému P je konečná neprázdná posloupnost přirozených čísel $i_1, i_2, \ldots, i_k$ taková, že $1 \leq i_j \leq n$ a

$$\alpha_{i_1} \alpha_{i_2} \ldots \alpha_{i_k} = \beta_{i_1} \beta_{i_2} \ldots \beta_{i_k}.$$

Postův korespondenční problém (PCP)

Postův korespondenční problém (PCP) je problém rozhodnout, zda má Postův systém P nějaké řešení.

$$PCP = \{\langle P\rangle \mid P \text{ je Postův systém, který má nějaké řešení}\}$$

Iniciální Postův korespondenční problém (inPCP) je problém rozhodnout, zda má Postův systém P nějaké řešení začínající číslem 1.

$$inPCP = \{\langle P\rangle \mid P \text{ je Postův systém, který má řešení začínající číslem 1}\}$$

Věta. PCP není rozhodnutelný.

Důkaz. Postupně ukážeme $ACC \leq_{\mathrm{m}} inPCP \leq_{\mathrm{m}} PCP$. $\square$

$\mathit{inPCP} \leq_{\mathbf{m}} \mathit{PCP}$

$ACC \leq_{\mathbf{m}} inPCP$